

РАЗВИТИЕ СИСТЕМЫ УПРАВЛЕНИЯ ПАКЕТАМИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИНКРЕМЕНТАЛЬНЫМ ОБНОВЛЕНИЕМ ФАЙЛОВ

Заикина А.В., Кузнецов М.А., Рогович В.И.

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия

DEVELOPMENT OF THE PACKAGE MANAGER WITH AN INCREMENTAL FILE UPDATE

Zaikina A.V., Kuznetsov M.A., Rogovich V.I.

Ural Federal University, Yekaterinburg, Russia

Annotation. The improvement of the package manager consists of creating a way to update software using a rolling hash-sum algorithm and an innovative update policy matrix, leading to a more efficient use of resources.

В настоящее время обновление пакетов программного обеспечения (ПО) повсеместно проводится не самым рациональным способом – путем загрузки нового пакета целиком.

Инкрементальное обновление файлов, то есть, обновление только изменившихся частей файла, позволит увеличить эффективность работы менеджера управления пакетами ПО, уменьшить затраты ресурсов на скачивание пакетов с сервера и время обновления пакетов в целом.

Авторами предложена архитектура системы обновлений, обладающая следующими возможностями:

1. инкрементальное обновление пакетов;
2. корректный перевод пакета на любую версию;
3. автоматический контроль зависимостей пакетов;
4. возможность установки нескольких версий пакета в систему пользователя;
5. запуск программы с нужными версиями зависимостей.

Инкрементального обновления файлов можно добиться путем использования алгоритма кольцевого хеша (rolling hash-sum) [1]. Алгоритм кольцевой хеш-суммы [1,2] позволяет с высокой скоростью находить отличающиеся части в двух массивах данных.

При обновлении программы требуется предусмотреть различные сценарии обновления файлов. Так, например, конфигурационные файлы нельзя просто заменить на новые, иначе пользователь при каждом обновлении будет терять все настройки. Такие файлы должны либо сохраняться неизменными, либо обновляться специальным скриптом. Кроме того, для корректной работы программы некоторые старые файлы (например, библиотеки) должны быть удалены из каталога установки.

Предлагаемый авторами алгоритм позволяет произвести обновление с учётом всех вышеперечисленных требований, выполняя загрузку только необходимых файлов.

Для перехода на новую версию предлагается создать список изменений, в котором каждый файл имеет определённую политику обновления: 1) keep (K) – оставить текущую версию файла или скачать новую, если файла не существует; 2) update (U) – полностью заменить текущий файл на новый или, используя алгоритм кольцевого хеша, обновить изменившиеся части; 3) scriptUpdate (S) – обновить текущий файл с помощью специального скрипта; 4) delete (D) – удалить текущий файл.

Для обновления пакета на несколько версий вперёд или назад предлагается использовать матрицу «сжатия» политик (сжатие политики U аналогично сжатию политики S):

Матрица «сжатия» политик обновления

		Новая версия			
		R_2	S_2	K_2	D_2
Старая версия	R_1	R	R	R	D
	S_1	R	$S_1 S_2$	S_1	D
	K_1	R	S_2	K	D
	D_1	R	-	-	D

В работе приводится обоснование данного алгоритма и вывод матрицы «сжатия» политик обновления.

1. Tridgell A, Mackerras P. The rsync algorithm (1996).
2. Pawan Bhadauria, “What is a rolling hash and when is it useful?”, <https://www.quora.com/What-is-a-rolling-hash-and-when-is-it-useful>